CLAIM AMENDMENTS

1. (original) A method of providing concurrent access to a resource object, the method comprising the computer-implemented steps of:

creating and storing a lock data structure for a particular resource object, the lock data structure comprising data indicative of values for a resource object identification, a lock type, and a version number related to a number of changes to the resource object since the lock data structure was generated;

receiving a request from a requesting process for a requested lock type for access to the particular resource object; and

determining whether to grant the request based on the requested lock type and the lock type in the lock data structure.

2. (currently amended) A method as recited in Claim 1, further comprising the step of:

if it is determined to grant the request, then

bringing the value of the lock type in the data structure into agreement with the lock type in the request;

generating a lock object having data indicative of the values for the resource object identification, the lock type and the version number from the lock data structure, and

returning the lock object to the requesting process.

3. (original) A method as recited in Claim 1, further comprising the steps of:

receiving a lock to be released having data indicative of values for the resource object identification and the lock type and the version number;

determining whether the data indicative of the value for the lock type in the lock to be released indicates an exclusive lock, and

if it is determined the data indicates the exclusive lock is to be released, then changing the value for the version number in the lock data structure based on the value of the version number in the lock to be released.

4. (currently amended) A method as recited in Claim ~~2~~ 1, wherein:

the lock data structure further comprises a reference number;

said step of ~~generating~~ creating a lock data structure further comprises setting the reference number to a predetermined initial value; and

said method further comprises, if it is determined to grant the request, then replacing the value of the reference number in the lock data structure with a sum of the value of the reference number in the lock data structure and a predetermined reference change value.

5. (original) A method as recited in Claim 4, further comprising the steps of:

receiving a lock to be released having data indicating the particular resource object;

determining whether the reference number substantially equals the predetermined initial value of the reference number; and

if it is determined the reference number does not substantially equal the predetermined initial value, then replacing the value of the reference number in the lock data structure with a difference substantially equal to the value of

the reference number in the lock data structure minus the predetermined reference change.

6. (original) A method as recited in Claim 5, further comprising, if it is determined the reference substantially equals the predetermined initial value, then deleting the lock data structure for the particular resource object.

7. (canceled)

8. (previously presented) A method of updating a resource object using optimistic locks, the method comprising the computer-implemented steps of:

receiving from a client process a request to update a particular resource object;

sending to a lock manager process a request for a first lock for access to the particular resource object, the request including data indicating an optimistic lock type;

receiving the first lock for access to the particular resource object, the first lock including data indicating the resource object, the optimistic lock type and a first value for a version number related to a number of changes to the resource object since the lock manager generated a lock data structure corresponding to the resource object; and

updating the resource object by

sending to a lock manager process a request for a second lock for access to the particular resource object, the request including data indicating the resource object identification and an exclusive lock type;

receiving the second lock for access to the particular resource object, the second lock including data indicating the resource object

identification, the exclusive lock type and a second value for the version number;

determining whether the second value for the version number substantially equals the first value for the version number; and

if the second value substantially equals the first value, then

committing an updated resource object to the resource, and

replacing the second value in the reference number in the second lock with a third value of the version number, the third value computed by adding the second value and a predetermined version change value.

9. (original) The method as recited in Claim 8, further comprising, if the second value does not substantially equal the first value, then sending a message to the client process, the message indicating that the resource object was not updated.

10. (previously presented) The method as recited in Claim 8, further comprising sending to the lock manager process a first release message to release the first lock.

11. (previously presented) The method as recited in Claim 10, further comprising sending to the lock manager process a second release message to release the second lock.

12. (previously presented) The method as recited in Claim 9, further comprising sending to the lock manager process a release message to release the second lock, the release message including data indicating the third value of the version number in the second lock and the

exclusive lock type, wherein the third value of the version number is used by the lock manager to replace the second value of the version number in the lock data structure.

13. (original) A computer-readable medium carrying one or more sequences of instructions for providing concurrent access to a resource object, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

creating and storing a lock data structure for a particular resource object, the lock data structure comprising data indicative of values for a resource object identification, a lock type, and a version number related to a number of changes to the resource object since the lock data structure was generated;

receiving a request from a requesting process for a requested lock type for access to the particular resource object; and

determining whether to grant the request based on the requested lock type and the lock type in the lock data structure.

14. (previously presented) A computer-readable medium carrying one or more sequences of instructions for updating a resource object, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

receiving from a client process a request to update a particular resource object;

sending to a lock manager process a request for a first lock for access to the particular resource object, the request including data indicating an optimistic lock type;

receiving the first lock for access to the particular resource object, the first lock including data indicating the resource object, the optimistic lock type and a

first value for a version number related to a number of changes to the resource object since the lock manager generated a lock data structure corresponding to the resource object; and

updating the resource object by

sending to a lock manager process a request for a second lock for access to the particular resource object, the request including data indicating the resource object identification and an exclusive lock type;

receiving the second lock for access to the particular resource object, the second lock including data indicating the resource object identification, the exclusive lock type and a second value for the version number;

determining whether the second value for the version number substantially equals the first value for the version number; and

if the second value substantially equals the first value, then

committing an updated resource object to the resource, and

replacing the second value in the reference number in the second lock with a third value of the version number, the third value computed by adding the second value and a predetermined version change value.

15. (original) An apparatus for providing concurrent access to a resource object, comprising:

a processor;

one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out the steps of:

creating and storing a lock data structure for a particular resource object, the lock data structure comprising data indicative of values for a resource object identification, a lock type, and a version number related to a number of changes to the resource object since the lock data structure was generated;

receiving a request from a requesting process for a requested lock type for access to the particular resource object; and

determining whether to grant the request based on the requested lock type and the lock type in the lock data structure.

16. (previously presented) An apparatus for updating a resource object, comprising:

a processor;

one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out the steps of:

receiving from a client process a request to update a particular resource object;

sending to a lock manager process a request for a first lock for access to the particular resource object, the request including data indicating an optimistic lock type;

receiving the first lock for access to the particular resource object, the first lock including data indicating the resource object, the optimistic lock type and a first value for a version number related to a number of

changes to the resource object since the lock manager generated a lock data structure corresponding to the resource object; and

updating the resource object by

sending to a lock manager process a request for a second lock for access to the particular resource object, the request including data indicating the resource object identification and an exclusive lock type;

receiving the second lock for access to the particular resource object, the second lock including data indicating the resource object identification, the exclusive lock type and a second value for the version number;

determining whether the second value for the version number substantially equals the first value for the version number; and

if the second value substantially equals the first value, then

committing an updated resource object to the resource, and

replacing the second value in the reference number in the second lock with a third value of the version number, the third value computed by adding the second value and a predetermined version change value.

17. (original) An apparatus for providing concurrent access to a resource object, comprising:

means for creating and storing a lock data structure for a particular resource object, the lock data structure comprising data indicative of values for a resource

object identification, a lock type, and a version number related to a number of changes to the resource object since the lock data structure was generated;

means for receiving a request from a requesting process for a requested lock type for access to the particular resource object; and

means for determining whether to grant the request based on the requested lock type and the lock type in the lock data structure.

18. (previously presented) An apparatus for updating a resource object, comprising:

a means for receiving from a client process a request to update a particular resource object;

a means for sending to a lock manager process a request for a first lock for access to the particular resource object, the request including data indicating an optimistic lock type;

a means for receiving the first lock for access to the particular resource object, the first lock including data indicating the resource object, the optimistic lock type and a first value for a version number related to a number of changes to the resource object since the lock manager generated a lock data structure corresponding to the resource object; and

a means for updating the resource object, including

a means for sending to a lock manager process a request for a second lock for access to the particular resource object, the request including data indicating the resource object identification and an exclusive lock type;

a means for receiving the second lock for access to the particular resource object, the second lock including data indicating the resource object

identification, the exclusive lock type and a second value for the version number;

a means for determining whether the second value for the version number substantially equals the first value for the version number;

a means for committing an updated resource object to the resource if the second value substantially equals the first value; and

a means for replacing the second value in the reference number in the second lock with a third value of the version number if the second value substantially equals the first value, the third value computed by adding the second value and a predetermined version change value.

19. (previously presented) The computer-readable medium as recited in Claim 14, wherein the instructions, when executed by one or more processors, cause the one or more processors to carry out the step of:

if the second value does not substantially equal the first value, then sending a message to the client process, the message indicating that the resource object was not updated.

20. (previously presented) The computer-readable medium as recited in Claim 14, wherein the instructions, when executed by one or more processors, cause the one or more processors to carry out the step of:

sending to the lock manager process a first release message to release the first lock.

21. (previously presented) The computer-readable medium as recited in Claim 20, wherein the instructions, when executed by one or more processors, cause the one or more processors to carry out the step of:

sending to the lock manager process a second release message to release the second lock.

22. (previously presented) The computer-readable medium as recited in Claim 19, wherein the instructions, when executed by one or more processors, cause the one or more processors to carry out the step of:

sending to the lock manager process a release message to release the second lock, the release message including data indicating the third value of the version number in the second lock and the exclusive lock type, wherein the third value of the version number is used by the lock manager to replace the second value of the version number in the lock data structure.

23. (previously presented) The apparatus as recited in Claim 16, wherein the instructions, when executed by one or more processors, cause the one or more processors to carry out the step of:

if the second value does not substantially equal the first value, then sending a message to the client process, the message indicating that the resource object was not updated.

24. (previously presented) The apparatus as recited in Claim 16, wherein the instructions, when executed by one or more processors, cause the one or more processors to carry out the step of:

sending to the lock manager process a first release message to release the first lock.

25. (previously presented) The apparatus as recited in Claim 24, wherein the instructions, when executed by one or more processors, cause the one or more processors to carry out the step of:

sending to the lock manager process a second release message to release the second lock.

26. (previously presented) The apparatus as recited in Claim 23, wherein the instructions, when executed by one or more processors, cause the one or more processors to carry out the step of:

sending to the lock manager process a release message to release the second lock, the release message including data indicating the third value of the version number in the second lock and the exclusive lock type, wherein the third value of the version number is used by the lock manager to replace the second value of the version number in the lock data structure.

27. (previously presented) The apparatus as recited in Claim 18, further comprising:

means for sending a message to the client process if the second value does not substantially equal the first value, the message indicating that the resource object was not updated.

28. (previously presented) The apparatus as recited in Claim 18, further comprising:
means for sending to the lock manager process a first release message to release the first lock.

29. (previously presented) The apparatus as recited in Claim 28, further comprising:
means for sending to the lock manager process a second release message to release the second lock.

30. (previously presented) The apparatus as recited in Claim 27, further comprising:
means for sending to the lock manager process a second release message to release the second lock, the second release message including data indicating the third value of the version number in the second lock and the exclusive lock type, wherein the third value of the version number is used by the lock manager to replace the second value of the version number in the lock data structure.